

Doxxygen_MPDASTARONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 12:08:25 2006

Contents

1	Doxygen_MPDAUTHENTICATION Class Index	1
1.1	Doxygen_MPDAUTHENTICATION Class List	1
2	Doxygen_MPDAUTHENTICATION File Index	3
2.1	Doxygen_MPDAUTHENTICATION File List	3
3	Doxygen_MPDAUTHENTICATION Class Documentation	5
3.1	trgClockSender Class Reference	5
3.2	trgHighVoltageSender Class Reference	14
4	Doxygen_MPDAUTHENTICATION File Documentation	25
4.1	trgClockDaemon.cc File Reference	25
4.2	trgClockSender.cc File Reference	26
4.3	trgClockSender.hh File Reference	27
4.4	trgClockSender_i.cc File Reference	28
4.5	trgHighVoltageDaemon.cc File Reference	29
4.6	trgHighVoltageSender.cc File Reference	30
4.7	trgHighVoltageSender.hh File Reference	31
4.8	trgHighVoltageSender_i.cc File Reference	32

Chapter 1

Doxxygen_MPDUSTARONLINE_DB Class Index

1.1 Doxygen_MPDUSTARONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

trgClockSender	5
trgHighVoltageSender	14

Chapter 2

Doxxygen_MPDAEMON_StarOnlineDB File Index

2.1 Doxygen_MPDAEMON_StarOnlineDB File List

Here is a list of all files with brief descriptions:

trgClockDaemon.cc	25
trgClockSender.cc	26
trgClockSender.hh	27
trgClockSender_i.cc	28
trgHighVoltageDaemon.cc	29
trgHighVoltageSender.cc	30
trgHighVoltageSender.hh	31
trgHighVoltageSender_i.cc	32

Chapter 3

Doxxygen_MP_D_STAR_ONLINE_DB Class Documentation

3.1 trgClockSender Class Reference

```
#include <trgClockSender.hh>
```

Public Member Functions

- `trgClockSender` (const char *localDir)
- virtual `~trgClockSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char *name, const char *value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char *fileName)
- virtual bool `updateDb` (const char *fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char * `readAny` ()
- bool `readVal` (char *&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)

Protected Attributes

- `trgClock previousVals` [NUM_DB_ROWS]
- `trgClock tempVals` [NUM_DB_ROWS]

- int `elementList` [NUM_DB_ROWS]
- `trgClock updateVals` [NUM_DB_ROWS]
- int `updateElements` [NUM_DB_ROWS]
- bool `mreadStatus`
- char `mline` [256]
- char `tmpline` [256]
- char * `ptr1`
- char * `ptr2`
- float `freqDriftLimit`

dito

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `trgClockSender::trgClockSender (const char * localDir)`

Definition at line 19 of file `trgClockSender.cc`.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("trgClock"); // setup the file I/O
24     initDataBase(); // database connections
25     initTable(); // table definitions
26
27 }
```

3.1.1.2 `virtual trgClockSender::~trgClockSender () [inline, virtual]`

Definition at line 41 of file `trgClockSender.hh`.

```
41 {};
```

3.1.2 Member Function Documentation

3.1.2.1 `bool trgClockSender::hasChanged (int rowNumber) [virtual]`

Definition at line 77 of file `trgClockSender_i.cc`.

```

77
78
79 trgClock* pre=&previousVals[rowNumber];
80 trgClock* cur=&tempVals[rowNumber];
81
82 if(fabs(pre->frequency-cur->frequency)>freqDriftLimit) return true;
83
84 return false;
85 }
```

3.1.2.2 void trgClockSender::initDataBase () [virtual]

Definition at line 75 of file trgClockSender.cc.

```

75
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("startrg","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbTrg;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.1.2.3 void trgClockSender::initQuery () [virtual]

Definition at line 39 of file trgClockSender_i.cc.

```

39
40 #define __METHOD__ "initQuery()"
41
42     ofstream to(queryFile);
43
44     if(!to.is_open()){
45         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
46         return;
47     }
48
49     to<<"read_freq"<<endl;
50
51     to.close();
52
53 #undef __METHOD__
54 }
```

3.1.2.4 void trgClockSender::initTable () [virtual]

Definition at line 30 of file trgClockSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("trgClock")))
35         sendMess("Could not find table=trgClock",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(trgClock));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(trgClock));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
```

```

46     sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47 }
48 memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50 unsigned int timestamp=time(NULL);
51 mgr->setRequestTime(timestamp);
52 if(mgr->fetchDbTable(table)){
53     trgClock* thv = (trgClock*)table->GetTable();
54     memcpy(previousVals,thv,nrows*sizeof(trgClock));
55 }
56
57 #undef __METHOD__
58 };

```

3.1.2.5 void **trgClockSender::initTags () [virtual]**

Definition at line 66 of file trgClockSender.cc.

```

66 {
67 /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("dephilli@bnl.gov");
69     setDomainName("trg");
70 }
71 }

```

3.1.2.6 bool **trgClockSender::loadUserControls (const char * name, const char * value) [virtual]**

Definition at line 20 of file trgClockSender_i.cc.

```

20 {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 */
26 if strstr(name,"freqDriftLimit")){
27     freqDriftLimit=atof(value);
28     sendMess("freqDriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
29     return true;
30 }
31
32
33 return false;
34 #undef __METHOD__
35 }

```

3.1.2.7 bool **trgClockSender::queryData () [virtual]**

Definition at line 91 of file trgClockSender.cc.

```

91 {
92 #define __METHOD__ "queryData()"
93
94 /*
95 * MORE THAN AN EXAMPLE....
96 * IF Standard SC-Query via "caGet" then,

```

```

97     * no need to change this method AT ALL
98     *
99     */
100
101    writeTime = (unsigned int)time(NULL);           //for database write time
102
103    /*char systemCmd[1024];
104    ostringstream scmd;
105    scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107    if(system((scmd.str()).c_str()))
108        return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110    return true;
111 #undef __METHOD__
112 }

```

3.1.2.8 char * trgClockSender::readAny ()

Definition at line 197 of file trgClockSender.cc.

```

197
198
199    strcpy(tmpLine,mLine);
200    ptr1=tmpLine;
201    ptr2=strtok(ptr1," ");
202    if(!ptr2) return ptr2;
203    ptr2=strtok(NULL," ");
204    return ptr2;
205 }

```

3.1.2.9 bool trgClockSender::readData (ifstream & from) [virtual]

Definition at line 59 of file trgClockSender_i.cc.

```

59
60 #define __METHOD__ "readData(ifstream)"
61
62 mReadStatus=true;
63 memset(tempVals,0,NUM_DB_ROWS*sizeof(trgClock));
64
65 #define __MISSING__ "**** Missing Data From Line = "
66
67 if(!from.getline(mLine,255) || !readVal(tempVals[0].frequency) ) mReadStatus=sendMess(__MISSING__,mLine);
68
69
70    from.close();
71 return true;
72 #undef __METHOD__
73 }

```

3.1.2.10 bool trgClockSender::readData (const char *fileName) [virtual]

Definition at line 116 of file trgClockSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"

```

```

118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from);    // user implemented file read
123 #undef __METHOD__
124 }
```

3.1.2.11 bool trgClockSender::readVal (long long & value)

Definition at line 270 of file trgClockSender.cc.

```

270
271
272     if(!readAny())return false;
273     char* store[256];
274     value= strtoll(ptr2,store,10);
275     if(strlen(*store)>0) return false; // value is not a number
276
277     return true;
278 };
```

3.1.2.12 bool trgClockSender::readVal (long & value)

Definition at line 259 of file trgClockSender.cc.

```

259
260
261     if(!readAny())return false;
262
263     char* store[256];
264     value= strtol(ptr2,store,10);
265     if(strlen(*store)>0) return false; // value is not a number
266
267     return true;
268 };
```

3.1.2.13 bool trgClockSender::readVal (int & value)

Definition at line 248 of file trgClockSender.cc.

```

248
249
250     if(!readAny()) return false;
251
252     char* store[256];
253     value=(int)strtol(ptr2,store,10);
254     if(strlen(*store)>0) return false; // value is not a number
255
256     return true;
257 };
```

3.1.2.14 bool trgClockSender::readVal (short & *value*)

Definition at line 237 of file trgClockSender.cc.

```
237
238
239     if(!readAny()) return false;
240
241     char* store[256];
242     value=(short)strtol(ptr2,store,10);
243     if(strlen(*store)>0) return false; // value is not a number
244
245     return true;
246 };
```

3.1.2.15 bool trgClockSender::readVal (double & *value*)

Definition at line 226 of file trgClockSender.cc.

```
226
227
228     if(!readAny())return false;
229
230     char* store[256];
231     value=strtod(ptr2,store);
232     if(strlen(*store)>0) return false; // value is not a number
233
234     return true;
235 };
```

3.1.2.16 bool trgClockSender::readVal (float & *value*)

Definition at line 214 of file trgClockSender.cc.

```
214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.1.2.17 bool trgClockSender::readVal (char *& *value*)

Definition at line 207 of file trgClockSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.1.2.18 bool trgClockSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file trgClockSender.cc.

```

127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     trgClock* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("trgClock");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.1.3 Member Data Documentation

3.1.3.1 int trgClockSender::elementList[NUM_DB_ROWS] [protected]

Definition at line 24 of file trgClockSender.hh.

3.1.3.2 float [trgClockSender::freqDriftLimit](#) [protected]

dito

Definition at line 35 of file trgClockSender.hh.

3.1.3.3 char [trgClockSender::mline\[256\]](#) [protected]

Definition at line 29 of file trgClockSender.hh.

3.1.3.4 bool [trgClockSender::mreadStatus](#) [protected]

Definition at line 28 of file trgClockSender.hh.

3.1.3.5 [trgClock](#) [trgClockSender::previousVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 22 of file trgClockSender.hh.

3.1.3.6 char* [trgClockSender::ptr1](#) [protected]

Definition at line 31 of file trgClockSender.hh.

3.1.3.7 char * [trgClockSender::ptr2](#) [protected]

Definition at line 31 of file trgClockSender.hh.

3.1.3.8 [trgClock](#) [trgClockSender::tempVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 23 of file trgClockSender.hh.

3.1.3.9 char [trgClockSender::tmpline\[256\]](#) [protected]

Definition at line 30 of file trgClockSender.hh.

3.1.3.10 int [trgClockSender::updateElements\[NUM_DB_ROWS\]](#) [protected]

Definition at line 26 of file trgClockSender.hh.

3.1.3.11 [trgClock](#) [trgClockSender::updateVals\[NUM_DB_ROWS\]](#) [protected]

Definition at line 25 of file trgClockSender.hh.

The documentation for this class was generated from the following files:

- [trgClockSender.hh](#)
- [trgClockSender.cc](#)
- [trgClockSender_i.cc](#)

3.2 trgHighVoltageSender Class Reference

```
#include <trgHighVoltageSender.hh>
```

Public Member Functions

- `trgHighVoltageSender (const char *localDir)`
- `virtual ~trgHighVoltageSender ()`
- `virtual void initTable ()`
- `virtual void initTags ()`
- `virtual void initDataBase ()`
- `virtual bool loadUserControls (const char *name, const char *value)`
- `virtual void initQuery ()`
- `virtual bool queryData ()`
- `virtual bool readData (const char *fileName)`
- `virtual bool updateDb (const char *fileName)`
- `virtual bool readData (ifstream &from)`
- `virtual bool hasChanged (int rowNumber)`
- `char * readAny ()`
- `bool readVal (char *&value)`
- `bool readVal (float &value)`
- `bool readVal (double &value)`
- `bool readVal (short &value)`
- `bool readVal (int &value)`
- `bool readVal (long &value)`
- `bool readVal (long long &value)`
- `bool nextLine (ifstream &from)`
- `void readError (int l, char *c, char *m)`

Protected Attributes

- `trgHighVoltage previousVals [NUM_DB_ROWS]`
- `trgHighVoltage tempVals [NUM_DB_ROWS]`
- `int elementList [NUM_DB_ROWS]`
- `trgHighVoltage updateVals [NUM_DB_ROWS]`
- `int updateElements [NUM_DB_ROWS]`
- `bool mreadStatus`
- `char mline [256]`
- `char tmpline [256]`
- `char * ptr1`
- `char * ptr2`
- `float driftLimit`

dito

3.2.1 Constructor & Destructor Documentation

3.2.1.1 **trgHighVoltageSender::trgHighVoltageSender (const char * *localDir*)**

Definition at line 19 of file trgHighVoltageSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("trgHighVoltage"); // setup the file I/O
24     initDataBase();        // database connections
25     initTable();           // table definitions
26
27 }
```

3.2.1.2 **virtual trgHighVoltageSender::~trgHighVoltageSender () [inline, virtual]**

Definition at line 41 of file trgHighVoltageSender.hh.

```
41 {};
```

3.2.2 Member Function Documentation

3.2.2.1 **bool trgHighVoltageSender::hasChanged (int *rowNumber*) [virtual]**

Definition at line 109 of file trgHighVoltageSender_i.cc.

```

109
110
111 trgHighVoltage* pre=&previousVals[rowNumber];
112 trgHighVoltage* cur=&tempVals[rowNumber];
113
114 if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
115 if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
116 if(fabs(pre->ch2Voltage-cur->ch2Voltage)>=driftLimit) return true;
117 if(fabs(pre->ch3Voltage-cur->ch3Voltage)>=driftLimit) return true;
118 if(fabs(pre->ch4Voltage-cur->ch4Voltage)>=driftLimit) return true;
119 if(fabs(pre->ch5Voltage-cur->ch5Voltage)>=driftLimit) return true;
120 if(fabs(pre->ch6Voltage-cur->ch6Voltage)>=driftLimit) return true;
121 if(fabs(pre->ch7Voltage-cur->ch7Voltage)>=driftLimit) return true;
122 if(fabs(pre->ch8Voltage-cur->ch8Voltage)>=driftLimit) return true;
123 if(fabs(pre->ch9Voltage-cur->ch9Voltage)>=driftLimit) return true;
124 if(fabs(pre->ch10Voltage-cur->ch10Voltage)>=driftLimit) return true;
125 if(fabs(pre->ch11Voltage-cur->ch11Voltage)>=driftLimit) return true;
126 if(fabs(pre->ch12Voltage-cur->ch12Voltage)>=driftLimit) return true;
127 if(fabs(pre->ch13Voltage-cur->ch13Voltage)>=driftLimit) return true;
128 if(fabs(pre->ch14Voltage-cur->ch14Voltage)>=driftLimit) return true;
129 if(fabs(pre->ch15Voltage-cur->ch15Voltage)>=driftLimit) return true;
130
131 /* example ... note -> change to any element requires db-update
132  * and thus returns true immediately
133  *
134  */
135 /*if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
136 /*if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
137 */
138 /*
139 */

}
```

```

140
141     return false;
142 }
```

3.2.2.2 void trgHighVoltageSender::initDataBase () [virtual]

Definition at line 75 of file trgHighVoltageSender.cc.

```

75
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbTrg;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.2.2.3 void trgHighVoltageSender::initQuery () [virtual]

Definition at line 37 of file trgHighVoltageSender_i.cc.

```

37
38 #define __METHOD__ "initQuery()"
39
40     ofstream to(queryFile);
41
42     if(!to.is_open()){
43         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
44         return;
45     }
46
47     for(int i=0;i<16;i++){
48         to<<"TRGhv:SUB_RD_V_1:<<i<<.E"<<endl;
49         to<<"TRGhv:SUB_RD_V_1:<<i<<.F"<<endl;
50         to<<"TRGhv:SUB_RD_V_1:<<i<<.G"<<endl;
51         to<<"TRGhv:SUB_RD_V_1:<<i<<.H"<<endl;
52         to<<"TRGhv:SUB_RD_V_1:<<i<<.I"<<endl;
53         to<<"TRGhv:SUB_RD_V_1:<<i<<.J"<<endl;
54         to<<"TRGhv:SUB_RD_V_1:<<i<<.K"<<endl;
55         to<<"TRGhv:SUB_RD_V_1:<<i<<.L"<<endl;
56         to<<"TRGhv:SUB_RD_V1_1:<<i<<.E"<<endl;
57         to<<"TRGhv:SUB_RD_V1_1:<<i<<.F"<<endl;
58         to<<"TRGhv:SUB_RD_V1_1:<<i<<.G"<<endl;
59         to<<"TRGhv:SUB_RD_V1_1:<<i<<.H"<<endl;
60         to<<"TRGhv:SUB_RD_V1_1:<<i<<.I"<<endl;
61         to<<"TRGhv:SUB_RD_V1_1:<<i<<.J"<<endl;
62         to<<"TRGhv:SUB_RD_V1_1:<<i<<.K"<<endl;
63         to<<"TRGhv:SUB_RD_V1_1:<<i<<.L"<<endl;
64     }
65     to.close();
66
67 #undef __METHOD__
68 }
```

3.2.2.4 void trgHighVoltageSender::initTable () [virtual]

Definition at line 30 of file trgHighVoltageSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("trgHighVoltage")))
35         sendMess("Could not find table=trgHighVoltage",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(trgHighVoltage));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(trgHighVoltage));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         trgHighVoltage* thv = (trgHighVoltage*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(trgHighVoltage));
55     }
56
57 #undef __METHOD__
58 };

```

3.2.2.5 void trgHighVoltageSender::initTags () [virtual]

Definition at line 66 of file trgHighVoltageSender.cc.

```

66
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("trg");
70
71 }

```

3.2.2.6 bool trgHighVoltageSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file trgHighVoltageSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 */
26     if(strstr(name,"driftLimit")){
27         driftLimit=atof(value);
28         sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);

```

```

29     return true;
30 }
31 return false;
32 #undef __METHOD__
33 }
```

3.2.2.7 bool trgHighVoltageSender::nextLine (ifstream & from) [inline]

Definition at line 74 of file trgHighVoltageSender.hh.

```

74
75     if(!from.getline(mline,255))return false;
76     return true;
77 }
```

3.2.2.8 bool trgHighVoltageSender::queryData () [virtual]

Definition at line 91 of file trgHighVoltageSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.2.2.9 char * trgHighVoltageSender::readAny ()

Definition at line 197 of file trgHighVoltageSender.cc.

```

197
198
199 strcpy(tmpline,mline);
200 ptr1=tmpline;
201 ptr2=strtok(ptr1," ");
202 if(!ptr2) return ptr2;
203 ptr2=strtok(NULL," ");
204 return ptr2;
205 }
```

3.2.2.10 bool trgHighVoltageSender::readData (ifstream & *from*) [virtual]

Definition at line 73 of file trgHighVoltageSender_i.cc.

```

73
74 #define __METHOD__ "readData(ifstream)"
75
76 mreadStatus=true;
77 memset(tempVals,0,NUM_DB_ROWS*sizeof(trgHighVoltage));
78
79 int l=__LINE__;
80 char* c=__CLASS__;
81 char* m=__METHOD__;
82
83 for(int i=0;i<NUM_DB_ROWS;i++) {
84     if(!nextLine(from) || !readVal(tempVals[i].ch0Voltage)) readError(l,c,m);
85     if(!nextLine(from) || !readVal(tempVals[i].ch1Voltage)) readError(l,c,m);
86     if(!nextLine(from) || !readVal(tempVals[i].ch2Voltage)) readError(l,c,m);
87     if(!nextLine(from) || !readVal(tempVals[i].ch3Voltage)) readError(l,c,m);
88     if(!nextLine(from) || !readVal(tempVals[i].ch4Voltage)) readError(l,c,m);
89     if(!nextLine(from) || !readVal(tempVals[i].ch5Voltage)) readError(l,c,m);
90     if(!nextLine(from) || !readVal(tempVals[i].ch6Voltage)) readError(l,c,m);
91     if(!nextLine(from) || !readVal(tempVals[i].ch7Voltage)) readError(l,c,m);
92     if(!nextLine(from) || !readVal(tempVals[i].ch8Voltage)) readError(l,c,m);
93     if(!nextLine(from) || !readVal(tempVals[i].ch9Voltage)) readError(l,c,m);
94     if(!nextLine(from) || !readVal(tempVals[i].ch10Voltage)) readError(l,c,m);
95     if(!nextLine(from) || !readVal(tempVals[i].ch11Voltage)) readError(l,c,m);
96     if(!nextLine(from) || !readVal(tempVals[i].ch12Voltage)) readError(l,c,m);
97     if(!nextLine(from) || !readVal(tempVals[i].ch13Voltage)) readError(l,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].ch14Voltage)) readError(l,c,m);
99     if(!nextLine(from) || !readVal(tempVals[i].ch15Voltage)) readError(l,c,m);
100 }
101
102   from.close();
103 return true;
104 #undef __METHOD__
105 }
```

3.2.2.11 bool trgHighVoltageSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file trgHighVoltageSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119   ifstream from(fileName);
120   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122   return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.2.2.12 void trgHighVoltageSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file trgHighVoltageSender.hh.

```

79
80   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }
```

3.2.2.13 bool trgHighVoltageSender::readVal (long long & value)

Definition at line 269 of file trgHighVoltageSender.cc.

```
269
270
271     if(!readAny())return false;
272     char* store[256];
273     value= strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

3.2.2.14 bool trgHighVoltageSender::readVal (long & value)

Definition at line 258 of file trgHighVoltageSender.cc.

```
258
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value= strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

3.2.2.15 bool trgHighVoltageSender::readVal (int & value)

Definition at line 247 of file trgHighVoltageSender.cc.

```
247
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

3.2.2.16 bool trgHighVoltageSender::readVal (short & value)

Definition at line 236 of file trgHighVoltageSender.cc.

```
236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

3.2.2.17 bool trgHighVoltageSender::readVal (double & *value*)

Definition at line 225 of file trgHighVoltageSender.cc.

```
225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

3.2.2.18 bool trgHighVoltageSender::readVal (float & *value*)

Definition at line 214 of file trgHighVoltageSender.cc.

```
214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.2.2.19 bool trgHighVoltageSender::readVal (char *& *value*)

Definition at line 207 of file trgHighVoltageSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211
212     return true;
213 }
```

3.2.2.20 bool trgHighVoltageSender::updateDb (const char **fileName*) [virtual]

Definition at line 127 of file trgHighVoltageSender.cc.

```
127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     trgHighVoltage* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137 }
```

```

138     numRows=NUM_DB_ROWS;
139     elements=elementList;
140     vals = tempVals;
141
142 } else {
143
144     for(int i=0; i<NUM_DB_ROWS; i++){
145         if(hasChanged(i)){
146             updateElements[numRows]=elementList[i];
147             updateVals[numRows] = tempVals[i];
148             previousVals[i]=tempVals[i];
149             numRows++;
150         }
151     }
152
153     elements = updateElements;
154     vals      = updateVals;
155 }
156
157 if(numRows==0) return sendMess(" No update required for",mibName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("trgHighVoltage");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168 if(!mgr->storeDbTable(dbTable)) {
169     addBackLog(writeTime);
170     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171 }
172
173 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175 return true;
176 #undef __METHOD__
177 }
```

3.2.3 Member Data Documentation

3.2.3.1 float **trgHighVoltageSender::driftLimit** [protected]

dito

Definition at line 35 of file trgHighVoltageSender.hh.

3.2.3.2 int **trgHighVoltageSender::elementList[NUM_DB_ROWS]** [protected]

Definition at line 24 of file trgHighVoltageSender.hh.

3.2.3.3 char **trgHighVoltageSender::mline[256]** [protected]

Definition at line 29 of file trgHighVoltageSender.hh.

3.2.3.4 bool [trgHighVoltageSender::mreadStatus](#) [protected]

Definition at line 28 of file [trgHighVoltageSender.hh](#).

3.2.3.5 trgHighVoltage [trgHighVoltageSender::previousVals\[NUM_DB_ROWS**\]](#) [protected]**

Definition at line 22 of file [trgHighVoltageSender.hh](#).

3.2.3.6 char* [trgHighVoltageSender::ptr1](#) [protected]

Definition at line 31 of file [trgHighVoltageSender.hh](#).

3.2.3.7 char * [trgHighVoltageSender::ptr2](#) [protected]

Definition at line 31 of file [trgHighVoltageSender.hh](#).

3.2.3.8 trgHighVoltage [trgHighVoltageSender::tempVals\[NUM_DB_ROWS**\]](#) [protected]**

Definition at line 23 of file [trgHighVoltageSender.hh](#).

3.2.3.9 char [trgHighVoltageSender::tmpline\[256\]](#) [protected]

Definition at line 30 of file [trgHighVoltageSender.hh](#).

3.2.3.10 int [trgHighVoltageSender::updateElements\[NUM_DB_ROWS**\]](#) [protected]**

Definition at line 26 of file [trgHighVoltageSender.hh](#).

3.2.3.11 trgHighVoltage [trgHighVoltageSender::updateVals\[NUM_DB_ROWS**\]](#) [protected]**

Definition at line 25 of file [trgHighVoltageSender.hh](#).

The documentation for this class was generated from the following files:

- [trgHighVoltageSender.hh](#)
- [trgHighVoltageSender.cc](#)
- [trgHighVoltageSender_i.cc](#)

Chapter 4

Doxxygen_MPDAEMON_Star_Online_Db File Documentation

4.1 trgClockDaemon.cc File Reference

```
#include "trgClockSender.hh"
#include <unistd.h>
```

Functions

- void [runSender](#) (const char **ldir*)

4.1.1 Function Documentation

4.1.1.1 void [runSender](#) (const char * *ldir*)

Definition at line 14 of file trgClockDaemon.cc.

```
14
15
16 CndDbSender* sender = new trgClockSender(ldir);
17
18 sender->initQuery();
19 for(;;) { //ever...
20     if(sender->hasBackLog())sender->cleanBackLog();
21     if(sender->queryData())sender->updateDb();
22     sleep(sender->sleepTime());
23 }
24
25 };
```

4.2 trgClockSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "trgClockSender.hh"
#include "StDbTable.h"
#include "trgClockSender_i.cc"
```

Defines

- #define **_CLASS_** "trgClockSender"
- #define **_METHOD_** "initTable()"
- #define **_METHOD_** "initDataBase()"
- #define **_METHOD_** "queryData()"
- #define **_METHOD_** "readData(fileName)"
- #define **_METHOD_** "updateDb(filename)"

4.2.1 Define Documentation

4.2.1.1 #define **_CLASS_** "trgClockSender"

Definition at line 17 of file trgClockSender.cc.

4.2.1.2 #define **_METHOD_** "updateDb(filename)"

4.2.1.3 #define **_METHOD_** "readData(fileName)"

4.2.1.4 #define **_METHOD_** "queryData()"

4.2.1.5 #define **_METHOD_** "initDataBase()"

4.2.1.6 #define **_METHOD_** "initTable()"

4.3 trgClockSender.hh File Reference

```
#include "CndDbSender.hh"
#include "trgClock.h"
```

Classes

- class [trgClockSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

4.3.1 Define Documentation

4.3.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file trgClockSender.hh.

4.4 trgClockSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"
- #define MISSING "**** Missing Data From Line = "

4.4.1 Define Documentation

4.4.1.1 #define METHOD "readData(ifstream)"

4.4.1.2 #define METHOD "initQuery()"

4.4.1.3 #define METHOD "loadUserControls(name,value)"

4.4.1.4 #define MISSING "**** Missing Data From Line = "

4.5 trgHighVoltageDaemon.cc File Reference

```
#include "trgHighVoltageSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char *ldir)

4.5.1 Function Documentation

4.5.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file trgHighVoltageDaemon.cc.

```
14
15
16     CndDbSender* sender = new trgHighVoltageSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.6 trgHighVoltageSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "trgHighVoltageSender.hh"
#include "StDbTable.h"
#include "trgHighVoltageSender_i.cc"
```

Defines

- #define __CLASS__ "trgHighVoltageSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.6.1 Define Documentation

4.6.1.1 #define __CLASS__ "trgHighVoltageSender"

Definition at line 17 of file trgHighVoltageSender.cc.

4.6.1.2 #define __METHOD__ "updateDb(filename)"

4.6.1.3 #define __METHOD__ "readData(fileName)"

4.6.1.4 #define __METHOD__ "queryData()"

4.6.1.5 #define __METHOD__ "initDataBase()"

4.6.1.6 #define __METHOD__ "initTable()"

4.7 trgHighVoltageSender.hh File Reference

```
#include "CndDbSender.hh"
#include "trgHighVoltage.h"
```

Classes

- class [trgHighVoltageSender](#)

Defines

- #define [NUM_DB_ROWS](#) 16

4.7.1 Define Documentation

4.7.1.1 #define NUM_DB_ROWS 16

Definition at line 16 of file trgHighVoltageSender.hh.

4.8 trgHighVoltageSender_i.cc File Reference

Defines

- #define METHOD "loadUserControls(name,value)"
- #define METHOD "initQuery()"
- #define METHOD "readData(ifstream)"

4.8.1 Define Documentation

4.8.1.1 #define METHOD "readData(ifstream)"

4.8.1.2 #define METHOD "initQuery()"

4.8.1.3 #define METHOD "loadUserControls(name,value)"

Index

~trgClockSender
 trgClockSender, 6

~trgHighVoltageSender
 trgHighVoltageSender, 15

__CLASS__
 trgClockSender.cc, 26
 trgHighVoltageSender.cc, 30

__METHOD__
 trgClockSender.cc, 26
 trgClockSender_i.cc, 28
 trgHighVoltageSender.cc, 30
 trgHighVoltageSender_i.cc, 32

__MISSING__
 trgClockSender_i.cc, 28

driftLimit
 trgHighVoltageSender, 22

elementList
 trgClockSender, 12
 trgHighVoltageSender, 22

freqDriftLimit
 trgClockSender, 12

hasChanged
 trgClockSender, 6
 trgHighVoltageSender, 15

initDataBase
 trgClockSender, 6
 trgHighVoltageSender, 16

initQuery
 trgClockSender, 7
 trgHighVoltageSender, 16

initTable
 trgClockSender, 7
 trgHighVoltageSender, 16

initTags
 trgClockSender, 8
 trgHighVoltageSender, 17

loadUserControls
 trgClockSender, 8
 trgHighVoltageSender, 17

mline
 trgClockSender, 13
 trgHighVoltageSender, 22

mreadStatus
 trgClockSender, 13
 trgHighVoltageSender, 22

nextLine
 trgHighVoltageSender, 18

NUM_DB_ROWS
 trgClockSender.hh, 27
 trgHighVoltageSender.hh, 31

previousVals
 trgClockSender, 13
 trgHighVoltageSender, 23

ptr1
 trgClockSender, 13
 trgHighVoltageSender, 23

ptr2
 trgClockSender, 13
 trgHighVoltageSender, 23

queryData
 trgClockSender, 8
 trgHighVoltageSender, 18

readAny
 trgClockSender, 9
 trgHighVoltageSender, 18

readData
 trgClockSender, 9
 trgHighVoltageSender, 18, 19

readError
 trgHighVoltageSender, 19

readVal
 trgClockSender, 10, 11
 trgHighVoltageSender, 19–21

runSender
 trgClockDaemon.cc, 25
 trgHighVoltageDaemon.cc, 29

tempVals
 trgClockSender, 13
 trgHighVoltageSender, 23

tmpline

trgClockSender, 13
 trgHighVoltageSender, 23
 trgClockDaemon.cc, 25
 trgClockDaemon.cc
 runSender, 25
 trgClockSender, 5
 trgClockSender, 6
 trgClockSender
 ~trgClockSender, 6
 elementList, 12
 freqDriftLimit, 12
 hasChanged, 6
 initDataBase, 6
 initQuery, 7
 initTable, 7
 initTags, 8
 loadUserControls, 8
 mline, 13
 mreadStatus, 13
 previousVals, 13
 ptr1, 13
 ptr2, 13
 queryData, 8
 readAny, 9
 readData, 9
 readVal, 10, 11
 tempVals, 13
 tmpline, 13
 trgClockSender, 6
 updateDb, 11
 updateElements, 13
 updateVals, 13
 trgClockSender.cc, 26
 trgClockSender.cc
 __CLASS__, 26
 __METHOD__, 26
 trgClockSender.hh, 27
 trgClockSender.hh
 NUM_DB_ROWS, 27
 trgClockSender_i.cc, 28
 trgClockSender_i.cc
 __METHOD__, 28
 __MISSING__, 28
 trgHighVoltageDaemon.cc, 29
 trgHighVoltageDaemon.cc
 runSender, 29
 trgHighVoltageSender, 14
 trgHighVoltageSender, 15
 trgHighVoltageSender
 ~trgHighVoltageSender, 15
 driftLimit, 22
 elementList, 22
 hasChanged, 15
 initDataBase, 16
 initQuery, 16
 initTable, 16
 initTags, 17
 loadUserControls, 17
 mline, 22
 mreadStatus, 22
 nextLine, 18
 previousVals, 23
 ptr1, 23
 ptr2, 23
 queryData, 18
 readAny, 18
 readData, 18, 19
 readError, 19
 readVal, 19–21
 tempVals, 23
 tmpline, 23
 trgHighVoltageSender, 15
 updateDb, 21
 updateElements, 23
 updateVals, 23
 trgHighVoltageSender.cc, 30
 trgHighVoltageSender.cc
 __CLASS__, 30
 __METHOD__, 30
 trgHighVoltageSender.hh, 31
 trgHighVoltageSender.hh
 NUM_DB_ROWS, 31
 trgHighVoltageSender_i.cc, 32
 trgHighVoltageSender_i.cc
 __METHOD__, 32
 updateDb
 trgClockSender, 11
 trgHighVoltageSender, 21
 updateElements
 trgClockSender, 13
 trgHighVoltageSender, 23
 updateVals
 trgClockSender, 13
 trgHighVoltageSender, 23